Position Paper
User Interaction with Multi-Robot Systems

David Kortenkamp, Debra Schreckenghost and
Cheryl Martin
NASA Johnson Space Center ER2
Metrica Inc./TRACLabs
Houston TX 77058

kortenkamp@jsc.nasa.gov

History?

- Historically, human-robot interaction research has focused on single humans interacting with single robots
- For many deployed systems there are *multiple* humans interacting with a single robot
- Little research on human interaction with multiple, autonomous robots
- Even less research on multiple humans interacting with multiple, autonomous robots
- *An architecture for distributed collaboration with multiple, autonomous robots*

Key Research Areas

- Adjustable autonomy
- Distributed task management
- Distributed user communication
- These areas are common to interaction with any autonomous system, not just robots
  - we can leverage other work

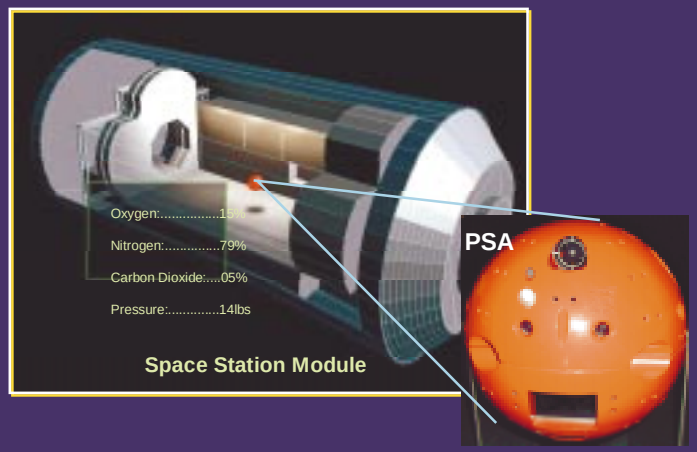
Robot Interaction Issues

- Teleoperation
  - Ability to step inside of robot to perform complicated tasks or to teach
- Multi-modal input
  - Vision, voice, tactile
- Physical interaction between robot and user
- These issues are specific to physical robots

User Interfaces for Robots

- Developer's interface
  - Intimate knowledge of internal workings
- "Ground" interface
  - Monitoring and troubleshooting
- End-user
  - Teleoperator
  - Team mate
  - Task manager

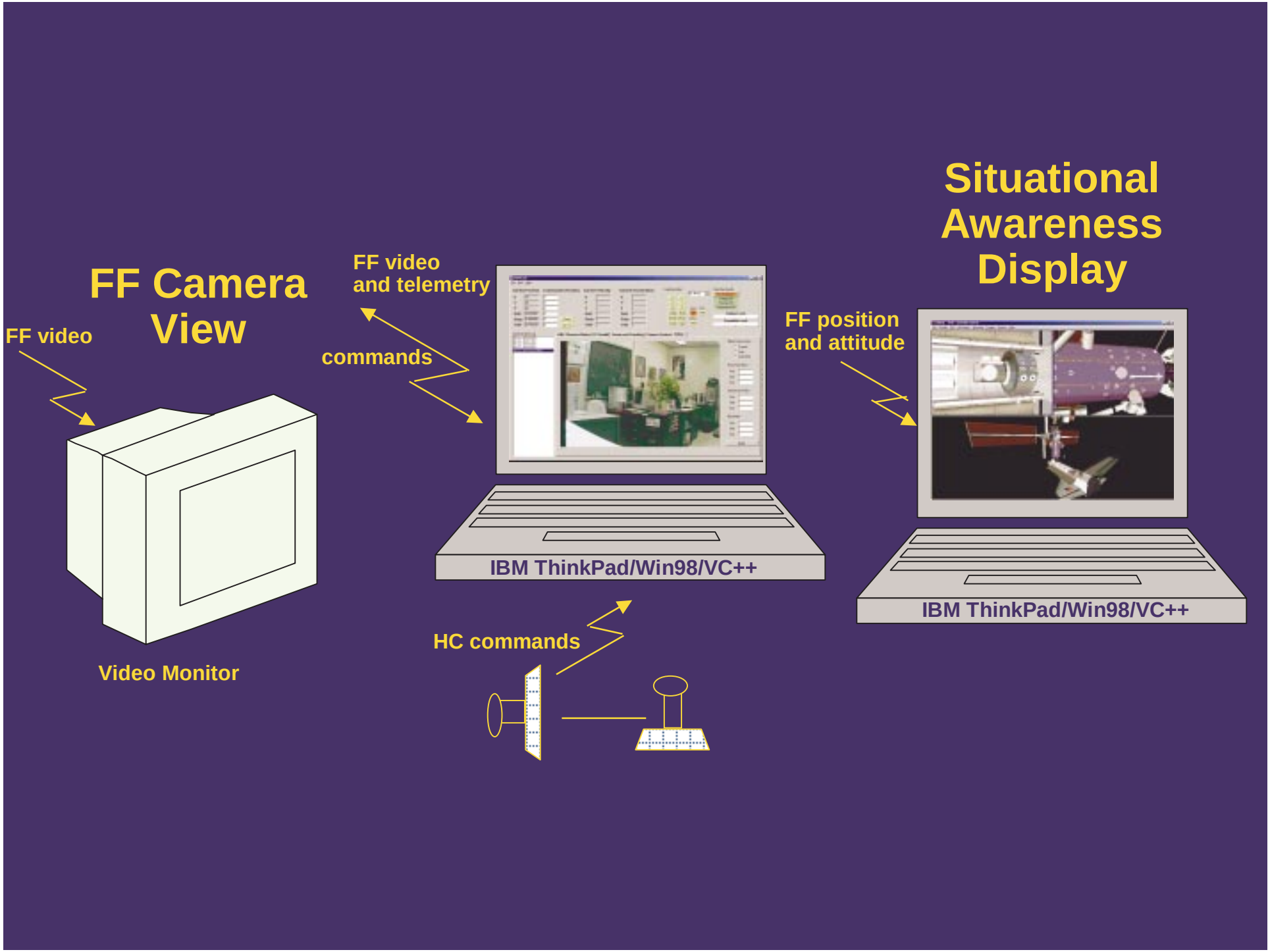
Some Examples



PSA

AERCam

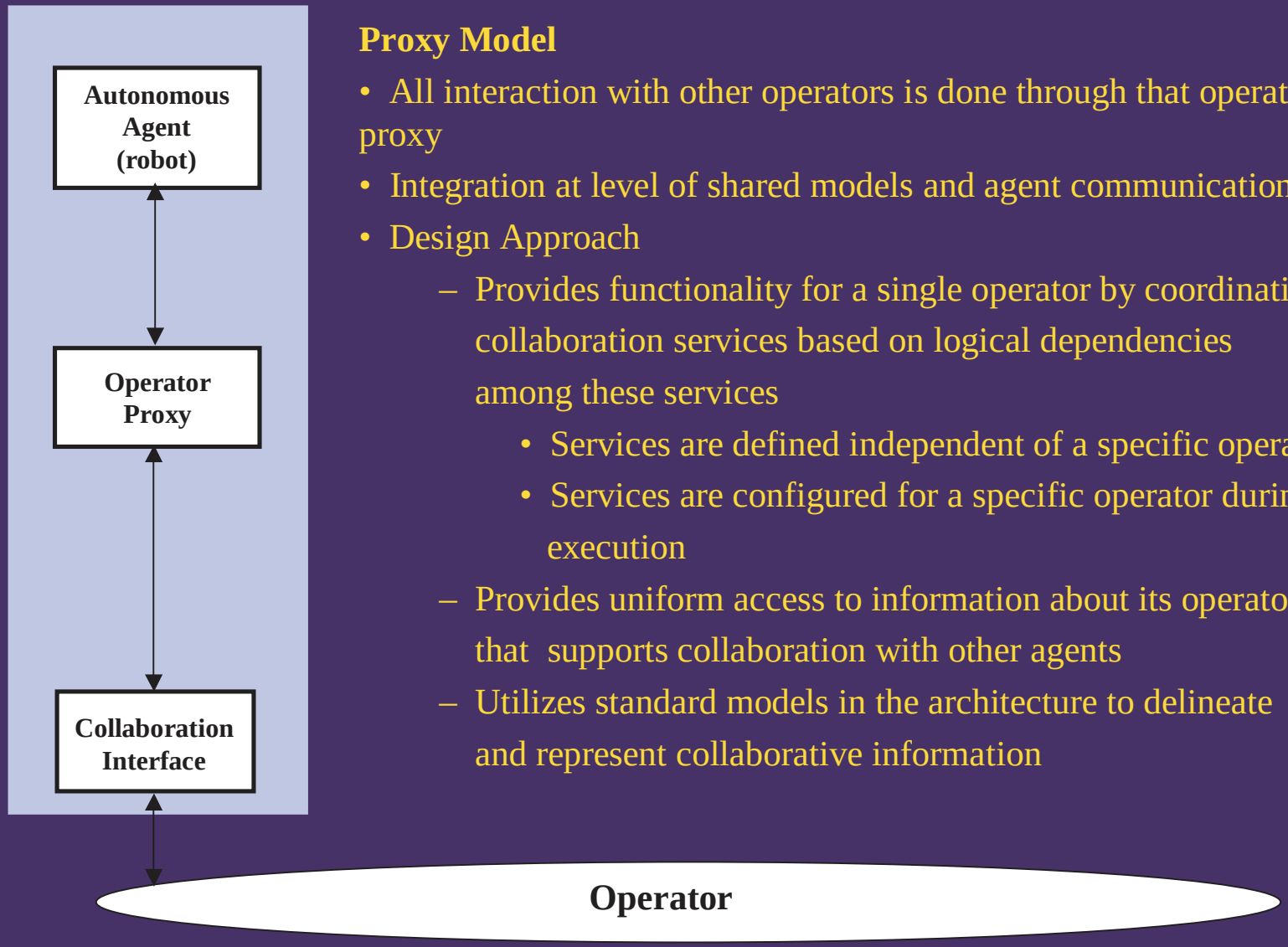
Robonaut


FF Camera View
FF video
Video Monitor
FF video and telemetry
commands
IBM ThinkPad/Win98/VC++
HC commands
Situational Awareness Display
FF position and attitude
IBM ThinkPad/Win98/VC++

Requirements for Distributed Interaction

- ***Visualization*** tools that characterize current situation
- ***Notification*** based on role, location, and preference
  - Primary operator should be the first notified when anomaly occurs and reminded to take action if required
  - Robot expert should be notified of all anomalies
  - Support is needed for locating available operator when primary unavailable
- ***Task management*** for robot operators
  - Notify/remind operators of manual tasks
  - Track completion of manual tasks
- ***Remote commanding*** strategies
  - Primary operator should be able to start, reconfigure, and stop automated control remotely
  - Commands authenticated and conflicts resolved for manual commands
  - Automatically ***adjust control autonomy*** for manual commands
- Assist operators in ***handling interruptions*** to normal activities

Model of Operator Interaction



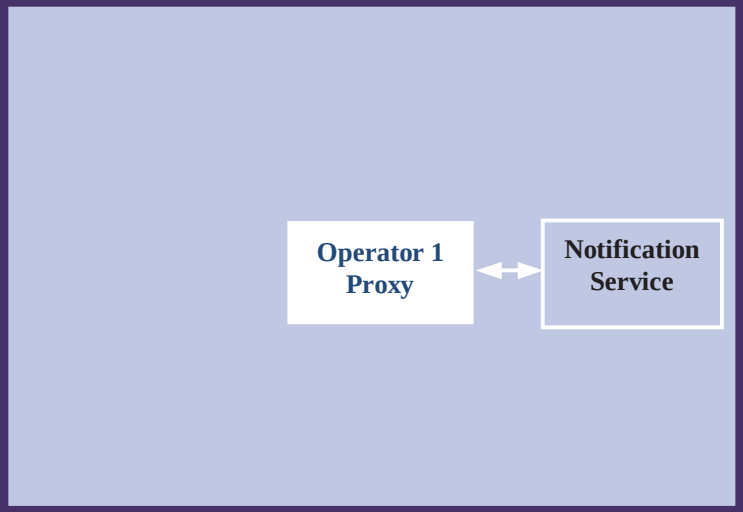
Proxy Model

- All interaction with other operators is done through that operator's proxy
- Integration at level of shared models and agent communication
- Design Approach
  - Provides functionality for a single operator by coordinating collaboration services based on logical dependencies among these services
    - Services are defined independent of a specific operator
    - Services are configured for a specific operator during execution
  - Provides uniform access to information about its operator that supports collaboration with other agents
  - Utilizes standard models in the architecture to delineate and represent collaborative information

Services for Operator Proxy

Notification Service

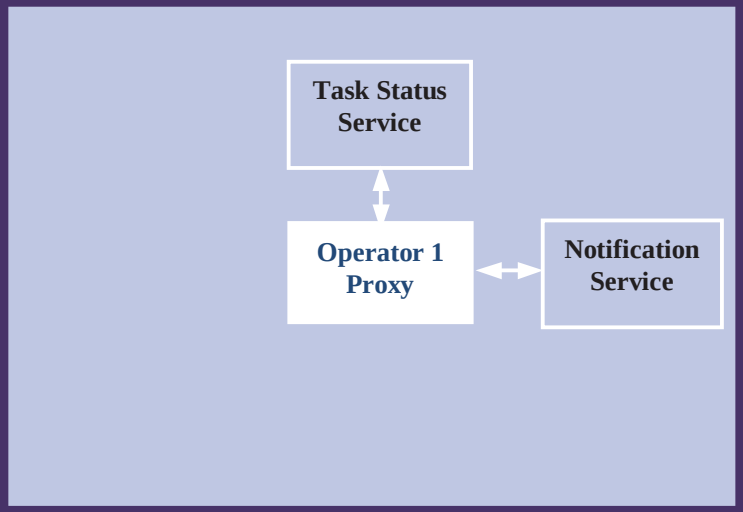
- Determines who to notify of an event and how to notify them
- Combines operator state (e.g., online vs offline), operator role, and operator notification preferences to do this
- Interesting events include environmental changes or control actions



Services for Operator Proxy

Task Status Service

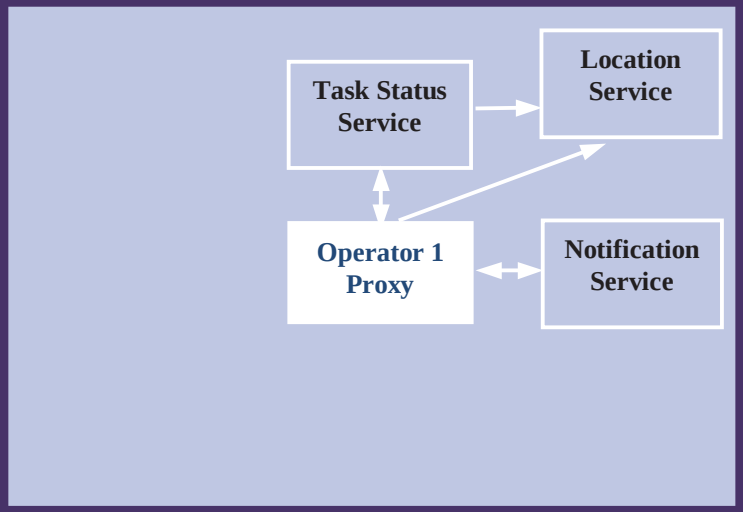
- Provides activity tracking and task management capabilities for both operators and robots
  - Assess the completion status of planned tasks
  - Notify the operator of scheduled tasks, including pending tasks and deadlines
  - Notify robots of the completion status of activities



Services for Operator Proxy

Location Service

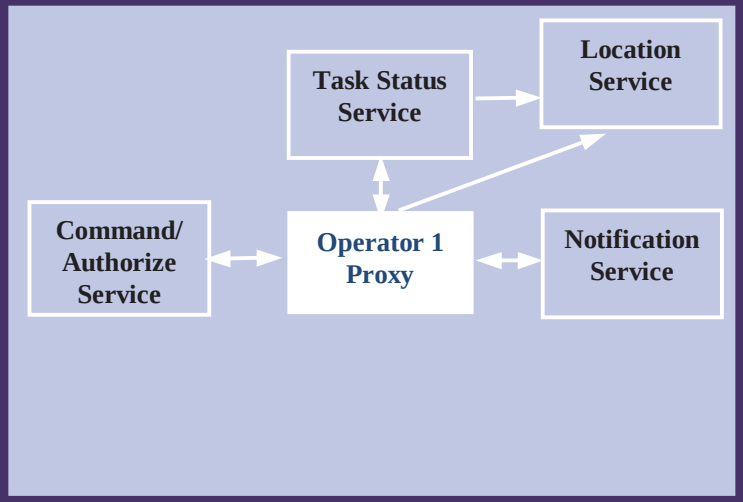
- Determines operator location as physical location, current computing platform, and whether operator is online or offline
- Provides operator location information for use in
  - Tracking the completion status of activities
  - Determining how to notify the operator of events
  - Customizing presentation of information



Services for Operator Proxy

Command and Authorization Service

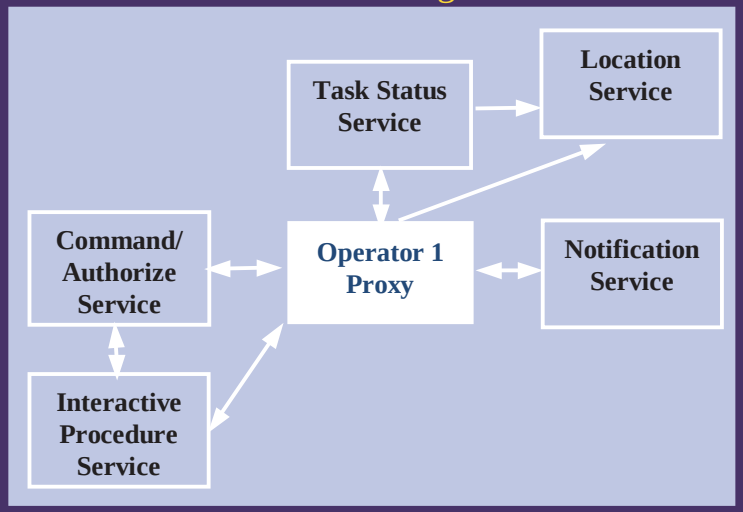
- Assists operators in remotely commanding robots
  - *Direct commanding* of novel tasks (e.g., teleoperation)
- Implements a concept for adjustable autonomy
  - Determines if the operator is authorized to command (i.e., access control)
  - Resolves authorization conflicts when more than one operator commands
  - Reconfigures both automation & user interface in preparation for commanding.



Services for Operator Proxy

Interactive Procedure Service

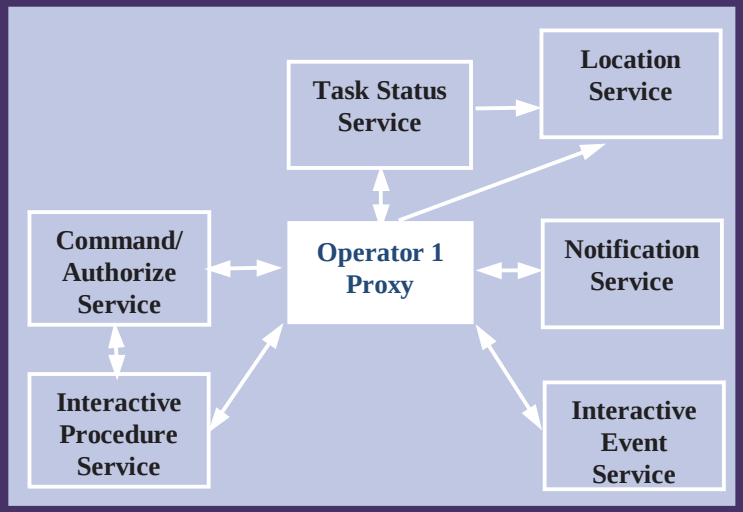
- Assists operator in temporarily modifying standard robot procedures
- Implements a shared discourse plan for Operator and Proxy to change procedure
- Guides structured modification of selected procedure
  - Change the steady state operating parameters (e.g., speed, acceleration)
  - Change to an alternative or backup sensors or actuators
  - Add probes to export information about the execution of control tasks
  - Temporarily disable automated commanding



Services for Operator Proxy

Interactive Event Service

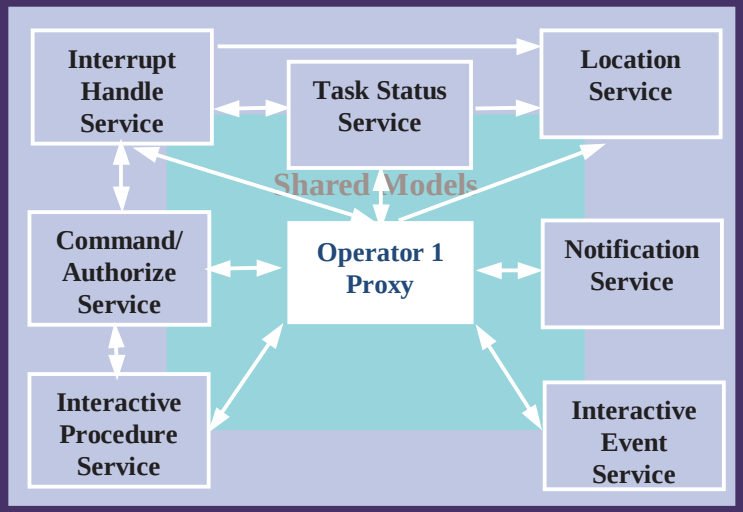
- Assists operator in interactively defining new operational events and controlling automated monitoring for these events
- Defined as data or action *probes* that are temporary event detectors and that generate information specific to the needs of a single operator
- Provide specialized operator interface to the Event Detection software and Interactive Procedure Service just described



Services for Operator Proxy

Interruption Handling Service

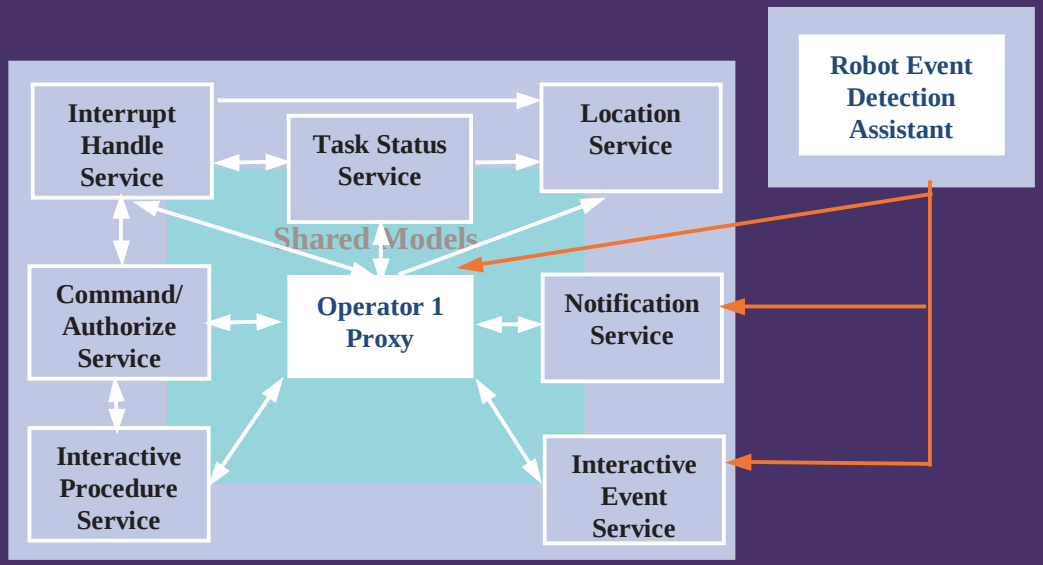
- Assists operator in responding to interruptions in normal operations
- Concepts for interruption handling
  - Determine if operator should be interrupted, and how intrusive interruption should be
  - Mark completion status of interrupted activities
  - Delegate an ongoing task by spawning a new automated task that "takes over" from operator
  - Assist operator in managing multiple, concurrent threads of activity



Control Assistant Software

Event Detection Assistant

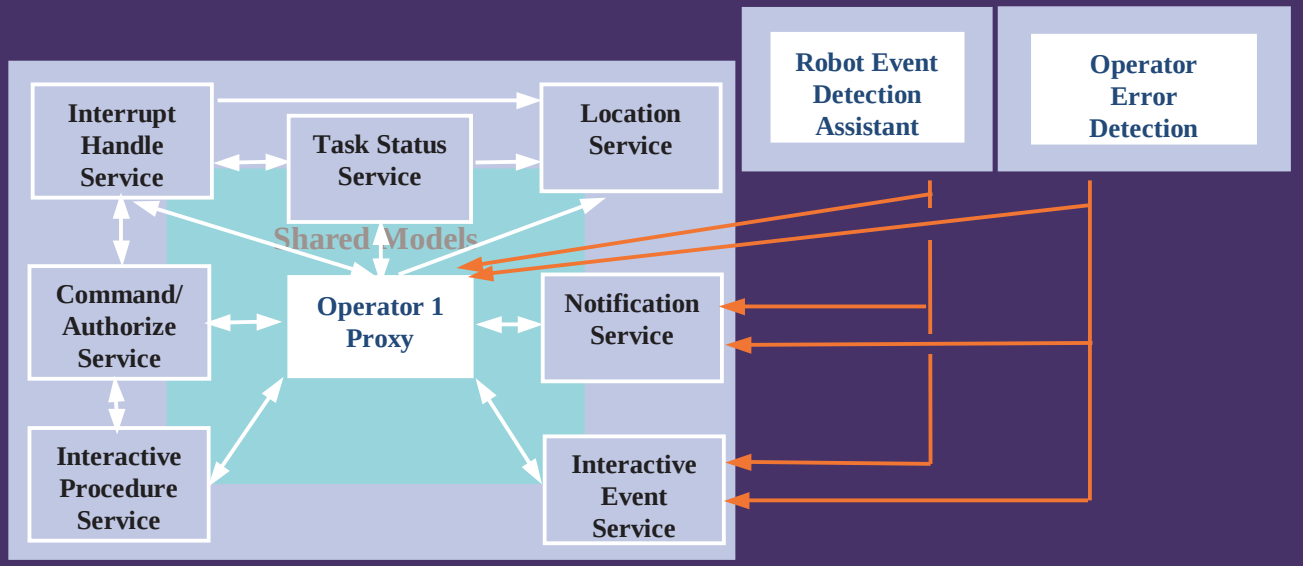
- Detects significant events (including anomalies) in robot systems
- Passes events to Notification Service of Operator Proxy
- Associates *simple events* with a single condition and a single time
- Detects *complex events* comprised of multiple simple events
- Ex: Detect stalled motor on mobile robot



Control Assistant Software

Operator Error Detection Assistant

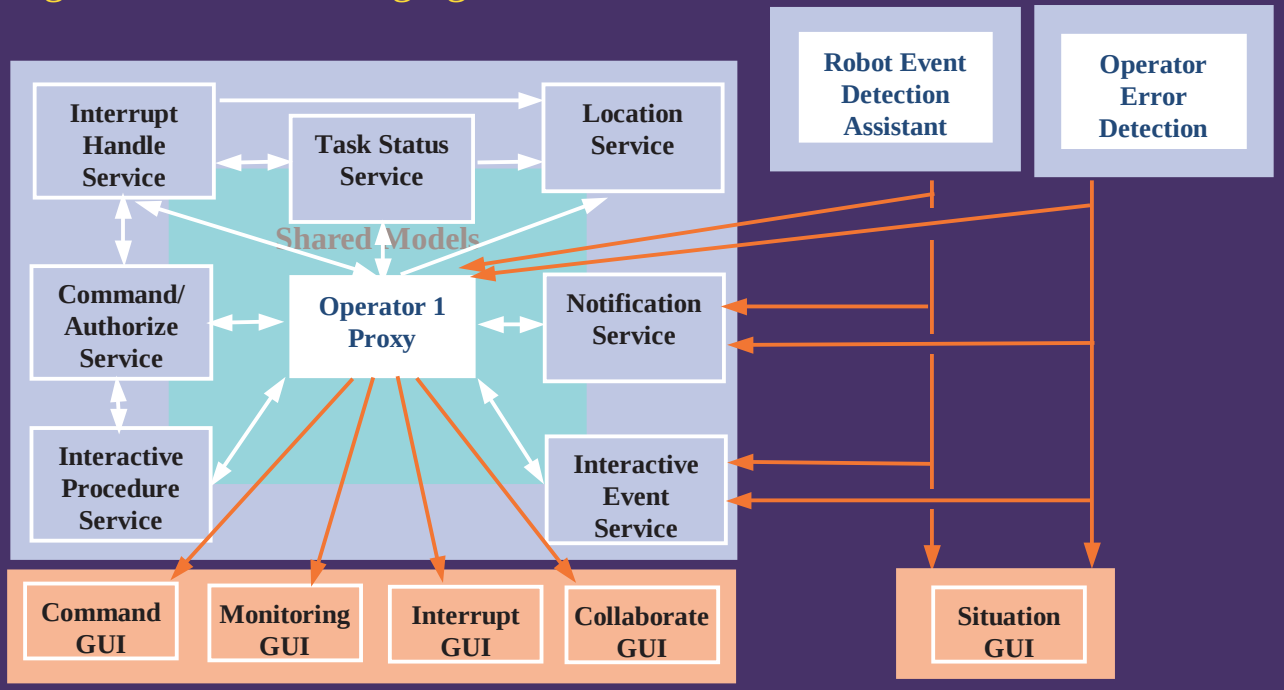
- Combines knowledge of operator tasks with knowledge of error pattern to detect and notify of operator error
- Example: Tool left in robot hand after teleoperation

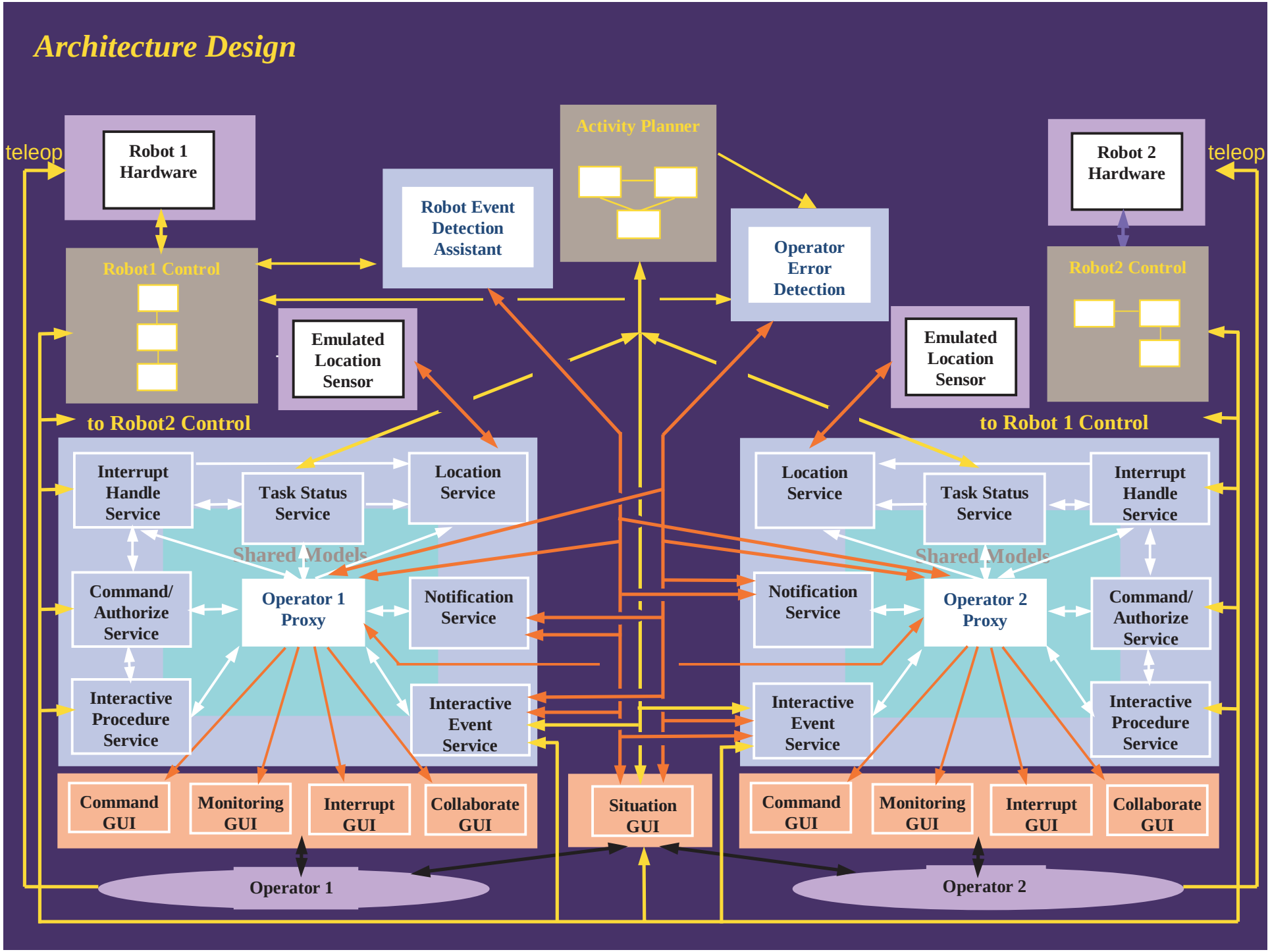


Collaborative User Interface

Collaborative User Interface

- Situation Interface for situation assessment;
- Monitoring Interface for event notification and task management
- Remote Commanding Interface; includes authorizing and issuing commands
- Collaboration Interface supporting operator-to-operator interaction
- Interruption Handling Interface for managing concurrent tasks




Architecture Design
teleop
Robot 1 Hardware
Robot1 Control
Activity Planner
Robot Event Detection Assistant
Operator Error Detection
Emulated Location Sensor
Robot 2 Hardware
Robot2 Control
teleop
Emulated Location Sensor
to Robot2 Control
to Robot 1 Control
Interrupt Handle Service
Task Status Service
Location Service
Shared Models
Command/ Authorize Service
Operator 1 Proxy
Notification Service
Interactive Procedure Service
Interactive Event Service
Location Service
Task Status Service
Interrupt Handle Service
Shared Models
Notification Service
Operator 2 Proxy
Command/ Authorize Service
Interactive Event Service
Interactive Procedure Service
Command GUI
Monitoring GUI
Interrupt GUI
Collaborate GUI
Situation GUI
Command GUI
Monitoring GUI
Interrupt GUI
Collaborate GUI
Operator 1
Operator 2

Status

- Project just started
- Proxy and services being implemented in Java and connected with CORBA
- First application is operator interaction with an autonomous controller (3T) for a life support system

Open Questions

- Proxy for robot?
- What belongs in robot control system and what belongs in operator interaction architecture?
- Support for teleoperation and teaching robots
- Scaling up to tens (hundreds?) of robots and many more users (ground)
- Will same architecture work for robots as different as AERCam (small DOF, small # of sensors) and Robonaut (high DOF, large # of sensors)? Where does robot specificity come in?
- Learning…